



***MedQuest Software***

# **MedQuest Analyzer Reference Manual**

***Version 3.03***

---

***April 27, 1999***

---

---

# TABLE OF CONTENTS

---

<b>OVERVIEW.....</b>	<b>1</b>
<i>Introduction.....</i>	<i>1</i>
<i>Conventions.....</i>	<i>1</i>
<b>QUICK START .....</b>	<b>3</b>
<i>Installation and Start Up .....</i>	<i>3</i>
Hardware .....	3
MedQuest Analyzer Components.....	3
MedQuest Analyzer Sample Analysis Module (AT2) .....	3
Install the MedQuest Analyzer .....	4
Help.....	4
<b>USING THE MEDQUEST ANALYZER .....</b>	<b>5</b>
<i>Analysis Overview.....</i>	<i>5</i>
Derived Variables.....	5
Indicators .....	6
Views.....	6
Graphs.....	6
Scripts.....	6
<b>MODULE MANAGEMENT .....</b>	<b>7</b>
<i>Select a Project File .....</i>	<i>7</i>
<i>Select a Module and Set the Module Paths.....</i>	<i>7</i>
<i>Log In .....</i>	<i>7</i>
User IDs.....	7
<i>Design/Analyze a Module .....</i>	<i>8</i>
<i>Analyze a Module.....</i>	<i>8</i>
<i>Rebuild a Module .....</i>	<i>8</i>
<i>Special Project Options .....</i>	<i>8</i>
Select Project Options .....	9
Module Location Options (Analysis Only).....	9
General Options .....	9
<i>Exit MedQuest Analyzer .....</i>	<i>10</i>
<b>DESIGN/CODE ANALYSIS.....</b>	<b>11</b>
<i>Analysis Design Window .....</i>	<i>11</i>
<i>Analyses.....</i>	<i>12</i>
Add an Analysis.....	12
Edit the Analysis.....	12
Delete the Analysis.....	12
Comment on the Analysis .....	13
<i>Build Derived Variables .....</i>	<i>13</i>
Add a Single Derived Variable .....	14
Examples of Single Derived Variables .....	14
Add a Derived Grid Variable .....	14
Examples of Derived Grid Variables .....	14
Edit Derived Variable Mode .....	15
Delete Derived Variable Mode .....	15

Edit Derived Equation Mode .....	15
View Derived Details Mode .....	16
Move Derived Variable Up .....	17
Move Derived Variable Down .....	17
Delete All Derived Variables .....	17
Query/Add Derived Variables .....	18
Exit Build a Derived Variable .....	18
<i>Calculate Derived Variables</i> .....	18
<i>Indicators</i> .....	18
Add/Edit an Indicator .....	19
Save Indicator as Derived Table .....	19
Indicator Type .....	19
Records to Include/Exclude .....	20
Variables to Display .....	20
Percent Breakdown .....	21
Group Records By .....	21
Order Records By .....	22
Variables to Include in Dynamic Queries .....	22
Manually Edit SQL .....	22
Other Functions .....	22
Delete Indicator/Graph/View/Script .....	22
Move Indicator/Graph/View/Script Up .....	22
Move Indicator/Graph/View/Script Down .....	22
<b>PERFORM ANALYSIS</b> .....	<b>23</b>
Analyze .....	23
View Table Results .....	23
Add/Edit View Logic .....	23
Build Dynamic Query .....	23
Change Logic for Records to Include/Exclude .....	24
Change Variables to Order By .....	24
Save View .....	25
Comment on the View .....	25
Export .....	25
Export Action .....	26
Destination Directory .....	26
File Formats .....	26
Options .....	27
Export .....	27
Name of Exported Files .....	28
Print View .....	28
Close Window .....	28
Exit .....	28
View Graph Results .....	28
Data to Graph .....	29
Variables Selection .....	29
Graph Settings .....	29
Retrieve a Graph .....	29
Save a Graph .....	30
Comment on the Graph .....	30
Export to BMP, WMF, or JPEG .....	30
Print Graph .....	30

Close Window .....	30
Exit .....	30
<i>View Report Results</i> .....	31
Append to Text Report File .....	31
<i>Build/Run Analysis Script</i> .....	31
Add Analysis Script .....	31
Edit Analysis Script .....	32
Run Analysis Script .....	32
<i>Capture Analysis to Web Page (HTML)</i> .....	33
Append to Text Report File .....	33
<b>SENDING ANALYSIS TO SOMEONE ELSE</b> .....	<b>34</b>
<i>Sending a Dictionary/Analysis</i> .....	34
<i>Sending Data</i> .....	34
<i>Sending Analysis Scripts</i> .....	34
<i>Sending Web Page (HTML) of Analysis</i> .....	34
<b>REPORTS</b> .....	<b>35</b>
<i>View/Print Reports</i> .....	35
<i>Selecting a Report</i> .....	35
<i>Dictionary Reports</i> .....	36
Dictionary Variables Report .....	36
Dictionary Detailed Report .....	36
Dictionary Comparison Report .....	36
Clinical Help Comparison Report .....	37
Data Analysis Variable Report .....	37
Data Entry Rules Report .....	37
<i>Analysis Reports</i> .....	37
Analysis Report .....	37
Derived Variable Report .....	38
Analysis Storyboard (HTML) Report .....	38
<b>HELP</b> .....	<b>39</b>
<i>General Help</i> .....	39
<i>Module Help</i> .....	39
<i>About Software</i> .....	39
<b>APPENDIX A</b> .....	<b>40</b>
<i>Design Functions</i> .....	40
Add a Variable .....	40
Variable Identification .....	40
Variable Type .....	40
Variable Options .....	44
<i>Query/Add Variables</i> .....	45
<b>GLOSSARY</b> .....	<b>47</b>
<i>List of Terms</i> .....	47

---

# OVERVIEW

---

---

## INTRODUCTION

---

The MedQuest Analysis Tool (**MedQuest Analyzer**) is a new addition to the MedQuest suite of software products. The complete set of MedQuest tools now enables you to: design a clinical data dictionary and data entry system; collect data; distribute data; and begin to analyze the collected data.

The **MedQuest Analyzer** provides the capability to create an Analysis, i.e., a set of logic that is organized into a single group, or query (SQL) of the data. For each project (module), several analyses may be created. These analyses enable you to organize the queries into different logical sets, e.g., one analysis may include processes of care, and another may include outcomes of care during a hospital stay.

In performing an analysis, you may want to create DERIVED variables (based on the collected data) and add these to the dictionary, e.g., create one derived variable that is True if there are any Heparin medications being given within 48 hours of admission; and another derived variable that is True if there are any symptoms that indicate the patient had an MI during the stay. Once you have created these variables, you can execute the logic to derive the values.

Several indicators may also be constructed for an analysis. An indicator is a clinical query against the data (both collected and derived) that returns a set of data, e.g., display all patients that had an MI within 48 hours of admission (based on the result of the derived MI variable being True). You can create this query in the **MedQuest Analyzer** by selecting variables that exist in the clinical data and creating an SQL statement. The results of the indicator can be presented in either a tabular (spreadsheet) or graphical format. You can also switch between the two representations for viewing. In addition, you can display up to 10 indicators on the screen at once for comparing results of different indicators.

During analysis, it may also be useful to dynamically modify specific parts of the indicator, e.g., modify the values of a variable (age > 65, age < 65) to determine the impact on the results. These modifications to the indicator can be saved as different VIEWS of the indicator.

As part of the MedQuest set of capabilities, this tool also enables you to send the dictionary (containing your variable descriptions and analysis queries) and data to other MedQuest users. In addition, the capability to print the results on the screen to a printer and also print the results to an HTML file, that can be browsed using the Internet browsing tools, is provided.



This analysis tool was created specifically to enable you to build queries of the MedQuest data. However, during the analysis process, the data may require more complex manipulation than this tool currently supports. When more sophisticated analysis is required, e.g., statistical analysis, this tool provides you with the capability to export the indicator and clinical data to a standard format (dBase III, IV) for use with other analysis software. In addition to the export, the analysis tool provides the capability to create an input statement so the data can easily be read into SAS or Stata.

---

## CONVENTIONS

---

Standard conventions have been adopted in the *MedQuest Analyzer Reference Manual* to facilitate locating and identifying desired information. These conventions and their definitions include the following:

<b>Bolded Title Case</b>	Windows, screen titles, program functions, variable names, and dialog boxes (e.g., the <b>Add Indicator</b> screen, etc.).
<b>Title Case and Italicized</b>	Text box or list box descriptions (e.g., the <i>Derived Variables (single) Selection</i> list box, etc.).
<b>Title Case</b>	Menu options (e.g., the General Help menu option, etc.).
<b>&lt;CAPITALS&gt;</b>	Menus, buttons, icons, and function keys (e.g., the HELP menu, the <CANCEL> button, the <NOTES> icon, the <T> key, etc.).
<b>Bolded</b>	User entered data (e.g., Enter the password <b>Analyzer</b> ).
	Notes Symbol. This symbol indicates that the user should take notice of the information.
	Advanced User's Symbol. This symbol indicates that the advanced user should take notice of the information.

---

# QUICK START

---

---

## INSTALLATION AND START UP

---

This section discusses the following topics:

### Hardware

---

To operate the MedQuest Analyzer Application, the following hardware and software is required:

- IBM or IBM-compatible personal computer with 80486/50 or higher processor
- VGA or VGA-compatible display monitor
- MS-DOS® version 5.0 or higher and Microsoft Windows™ version 3.1, '95, '98 or NT
- Microsoft or Microsoft-compatible mouse or pointing device
- Hard disk with a minimum of 12 megabytes of space
- Eight megabytes of RAM (sixteen are recommended)

### MedQuest Analyzer Components

---

The MedQuest Analyzer software components consist of:

- **MedQuest Analyzer (ANL).** This diskette contains the MedQuest Analyzer application.
- **Libraries.** This diskette contains a number of additional utility files required by the MedQuest Analyzer.
- **Medications.** The medications and abbreviation databases are contained on this diskette.
- **Medications '98.** The Multum Medisource™ Lexicon Medications database is contained on this diskette.
- **ICD9.** A database with ICD-9-CM diagnosis and procedure codes is contained on this diskette.

To install, reinstall, and uninstall the MedQuest Analyzer software, follow the instructions described in the following sections.

### MedQuest Analyzer Sample Analysis Module (AT2)

---

**AT2** has been included as a sample module on the MedQuest Analyzer diskette. When the MedQuest Analyzer is installed, the AT2 module will be included in the same directory. To access the module, type the three letter acronym (AT2) in the *Module Acronym* box, type a description (i.e., **AT2** Sample Analysis Module) in the *Module Description* box, and complete the paths for the databases (i.e., dictionary, datastore, medications, ICD9, abbreviations, and employee).

The **AT2** module includes: Derived Variables, Indicators, Graphs, Views, and Scripts.

Credit is given to the Texas Medical Foundation (TX PRO), in particular, Lori Lee and Andi Gillentine for their assistance in developing a sample module.

## Install the MedQuest Analyzer

---

The installation of the MedQuest Analyzer application and other required components are explained in detail in the README.TXT file found on the installation diskette.

## Help

---

**General Help** provides general MedQuest Analyzer application help. **Module Help** is available if you have compiled the clinical help entered in your module (in MedQuest). (XXXDICT.HLP where XXX is the module acronym.) **About (MedQuest Analyzer) Help** identifies software and dictionary version information.

You can also open context sensitive help from any window on which you are working by pressing the <F1> key.



---

# USING THE MEDQUEST ANALYZER

---

---

## ANALYSIS OVERVIEW

---

The MedQuest suite of software tools provides you with several capabilities for collecting data and performing an analysis on these data. These capabilities are briefly described below and can be used as a template for the use of the analysis tool.

**BUILD A DATA DICTIONARY.** The **MedQuest** design tool lets you define all of the data variables that you want to collect and also design the screens that will be used to collect the data. All of this information is saved in the **MedQuest** dictionary (XXXDICT.MDB where XXX is the module acronym). By saving this information in this dictionary (which is really a database) you can easily perform version control (compare different versions of the dictionary to see changes) and reuse different variables built by other users.

**COLLECT DATA.** The **MedQuest** data entry engine allows you to collect data once you have built the dictionary and data entry screens. This system allows for data collection in a multi-user or standalone environment. The data are saved in the clinical datastore database (XXX.MDB where XXX is the module acronym).

**QUALITY CONTROL DATA.** As you collect the data, you can perform quality control (if you are doing retrospective data collection) by using the **MedQuest Quality (IQC)** utility. This tool allows for the same case to be abstracted by different users and has a mechanism for comparing the results and pointing out quality control problems. In addition, you can build a series of quality control queries using the **MedQuest Analyzer** (or other tools such as Microsoft Access) that can identify quality control problems by searching for certain patterns in the data (e.g., data out of range).

**DATA ANALYSIS.** An analysis is a group of related items used to answer questions about the clinical data. Once the data are collected, you want to begin analyzing them. This analysis process is typically an open ended approach that may take different paths depending on the information collected, e.g., you may perform statistical analysis on the data or use a commercial reporting tool to produce a set of reports.

Often, the analysis process involves the construction of a set of questions or queries of the data. The result of this analysis produces a subset of the raw data that can be easily represented. This process of query building is what the analysis tool was built to support since it is a relatively straightforward and simple process compared to more complex steps of analysis.

The MedQuest Analysis Tool lets you accomplish this activity by building **Derived Variables, Indicators, Graphs, Views and Scripts**. These analysis objects and the methods for creating and using them during analysis are described below:

---

## Derived Variables

---

Often in analysis, you may find it necessary to use the raw data to infer a given result, i.e., whether a given clinical event such as an MI took place. The **MedQuest Analysis Tool** lets you do this by building a derived variable. You can build a derived variable by building an equation that describes the result. For example, you can build a derived rule that says "If ... and ... and ... then MI=true."

In the analysis process, this stage of development is laying the foundation for performing more complex queries by breaking the parts of the complex queries into smaller pieces.

## Indicators

---

In the MedQuest Analyzer, you can build your query of the data by creating an **Indicator**. When you create (or edit) an indicator, you are selecting the variables that you want in your query. The analysis tool will generate an SQL statement after the indicator (query) building template is completed.

To build a query (in equation form), options are provided to describe the records to be included or excluded, the variables to be displayed, the variables to be grouped by (and filtered for), and the variables to be ordered by. In addition, the capability is provided to identify variables to be included in dynamic queries during the analysis, e.g., you may want to vary the diagnosis code that is included (or excluded) as the data are analyzed.

An indicator will produce a data table that is displayed and that can also be graphically represented.

## Views

---

As you are analyzing the results of an indicator (the SQL query has been executed), you may want to make "minor" variations in the logic to see how the results change. For example, you may want to analyze the change in results for age range < 65 compared to age range > 70. In the MedQuest Analyzer, this process of modifying the indicator (query) logic is called creating a view. The MedQuest Analyzer enables you to dynamically modify the records to be included/excluded and also the order of the data. When you observe a view that you want to keep, you can save this view and it will be included under the indicator.

## Graphs

---

When working with an INDICATOR or a VIEW, it is often useful to present the data in a graphical format. The MedQuest Analyzer lets you create any number of graphs (based on the variations in the data) for both the indicator logic or view logic.

## Scripts

---

In performing an analysis, you will be creating and modifying the three basic analysis objects: Indicator; View; and Graph. You may want to organize these different sets of objects into different sequences to present or explain the results of your analysis. This sequencing can be accomplished using the SCRIPT object. In this object type, you can add any indicator, view or graph that is a part of the given analysis to the script and then allow the user to "play back" the script in the sequence you constructed.

The MedQuest Analysis Tool provides you with the capability to begin building a "data warehouse" of clinical data that can be queried and the results presented in tabular or graphical format. These data can then be used with other statistical packages to perform more complex analysis. The MedQuest Analysis Tool supports this next step by enabling you to export the data you have constructed for use with other analytical software.

---

# MODULE MANAGEMENT

---

---

## SELECT A PROJECT FILE

---

To begin using the MedQuest Analyzer, set the path to a valid **Project Database** (FILES.MDB) by entering the path in the box on the screen. The Project Database contains the locations of the data dictionary, clinical data, Medications/ICD9/Abbreviations Databases, and Employee Database of the modules in the project.

---

## SELECT A MODULE AND SET THE MODULE PATHS

---

Select a desired module from the *List of Modules* list box or type the acronym that represents the desired module in the *Module Acronym* text box. After the module has been selected, indicate the paths to the following databases:

<b>Path/File Name to Dictionary Database</b>	Indicates the path and file name for the module dictionary file (e.g., CCPDICT.MDB or any valid file name). This is the file where all of your design information and clinical help are saved.
<b>Path to Datastore Database</b>	Indicates the path for the module datastore file (e.g., CCP.MDB). This is the file where all of the data that were entered are saved.
<b>Path to Medications/ICD9/Abbreviations (MEDS.MDB, MMEDS.MDB, ICD9.MDB and ABB.MDB) Databases</b>	Indicates the path for the Medications database (MEDS.MDB/MMEDS.MDB), the ICD9 database (ICD9.MDB) and the Abbreviations database (ABB.MDB).
<b>Path to Users Database</b>	Indicates the path for the module Employee database (EMPLOYEE.MDB).



The MedQuest Analyzer validates whether the path to each file is in the location indicated. Valid information is displayed in black; invalid information is displayed in red.



The MedQuest Analyzer tool is currently not compatible with ODBC. If a module is using an ODBC data source in which to collect data, the ODBC option must be deselected before using the MedQuest Analyzer tool to perform analyses.

---

## LOG IN

---

The user can log in to the MedQuest Analyzer by entering the user identification and password in the *User Name* and *Password* text boxes, respectively.

---

## User IDs

---

Use the following log in identifications and passwords for the MedQuest Analyzer Application:

- • User log in identification **ANALYZER** and password **ANALYZER**.
- • User log in identifications **ID1** through **ID100** and passwords **ID1** through **ID100**, correspondingly.

The user log in identification and password can be added, modified, or deleted by using the MedQuest Manager (SMS) Application.

---

## DESIGN/ANALYZE A MODULE

---

Once a module has been selected, click on the <DESIGN/ANALYZE> button. If this is the first time this module has been loaded into the **MedQuest Analyzer**, the analysis tables will be added and then the **MedQuest Analysis Tool** screen will be displayed so you may begin the analysis process. The functions available in the **Design/Analyze** process include: adding, editing, and deleting analyses, indicators, scripts, and derived variables; viewing/printing reports; querying variables; and reviewing help.

---

## ANALYZE A MODULE

---

If you select a module for which you have already created analyses and you only want to perform analysis, click on the <ANALYZE> button. The **MedQuest Analysis Tool** screen will be displayed. The only functions available are to analyze, view/print reports, query variables, and review help.

---

## REBUILD A MODULE

---

When you first come into a module to perform an analysis, you can load the analysis without executing the rebuild function. If the MedQuest data dictionary does not match the data structure, you will be prompted to rebuild. If this occurs, then you have the wrong version of either the data dictionary and/or the datastore database.

You can use the **Rebuild** function to rebuild the datastore database. As in the MedQuest data entry engine, when you rebuild, the datastore database is backed up and rebuilt and you are given the option of "copying" the existing data into the new rebuild tables.

Even when the datastore and data dictionaries are correctly matched, you may find it useful at times to execute the rebuild to clear the derived variable data tables. WHEN YOU EXECUTE THE REBUILD FUNCTION, THE DERIVED VARIABLE TABLES WILL BE CLEARED. This will force you to recreate the derived variable results by recalculating them using the function provided.

---

## SPECIAL PROJECT OPTIONS

---

If you are going to distribute your module to another site and only wish to send them the data entry engine (no design allowed), you might want to take advantage of some of the following features.

## Select Project Options

---

Once you have selected a valid **Project Database** (FILES.MDB), you can edit the Project Options by clicking on the <PROJECT OPTIONS> button on the **Log In/Manage Module** screen. This will display the **Edit Project Options** screen.

When you distribute an application to a user, you may want to distribute only the Data Entry Engine and/or the Analysis Tool with certain restrictions. There are several options available for each of these applications.

### NOTE:

These options are saved in the Project Database: C:\MDQ400\FILES.MDB (your directory). You should distribute this FILES.MDB (along with you dictionary and datastore databases) to the designated users with the MedQuest Installer, Data Entry Engine, Libraries, Medications, and, if applicable, the Analysis, Analysis Libraries, and the ICD9 disks.

### NOTE:

After making changes, you must restart the application for the new FILES.MDB.

## Module Location Options (Analysis Only)

---

**USE DEFAULT.** The user has access to all the files listed in the FILES.MDB and can change any of the database paths.

**ALLOW MODULE PATHS TO BE CHANGED IN THE FILES.MDB AS IN THE MEDQUEST DESIGNER.** The user has the capability to change the database paths. Choose this option if you have only given the user **Analyze** capabilities.

**SEARCH LOCAL DIRECTORY FOR ALL EXISTING MODULES. USE FILES.MDB.** The FILES.MDB will list all of the modules located in your local directory. The user will only have access to this directory and will not be able to change the database paths.

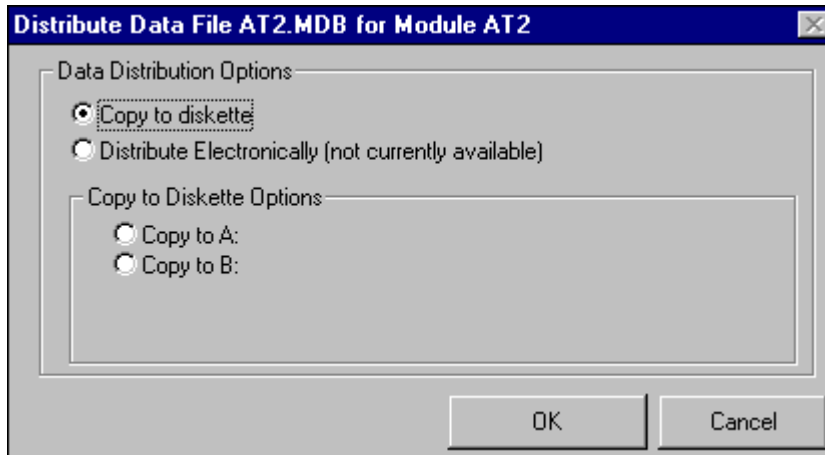
### NOTE:

This assumes that the dictionary, datastore, meds, and employee databases are all in the directory where the ANALYSIS.EXE resides.

## General Options

---

**DISPLAY DATA DISTRIBUTION BUTTON.** This option will enable the <DISTRIBUTE DATA> button on the **Log In/Manage Module** screen. Clicking on the <DISTRIBUTE DATA> button will display the *Distribution Data File* popup box.



Currently, the only available option is Copy to Diskette. The Distribute Electronically option will be available in a future release. By default, the Copy to Diskette option is enabled. Select the directory location from the *Copy to Diskette Options* and click on the <OK> button to copy your datastore database to diskette.

**DISPLAY ONLY PROJECT LIST/DESCRIPTION (DATA ENTRY ENGINE/ANALYSIS).** This option will display the **Log In/Manage Modules** screen without displaying the paths to the database files.

This feature can only be utilized for the Analyzer application when the Allow Analysis Only (No Design) option has also been selected.

**ANALYSIS TOOL: ALLOW ANALYSIS ONLY (NO DESIGN).** This option will provide the user with access to the **Analyze** feature only.

---

## EXIT MEDQUEST ANALYZER

---

Exit the MedQuest Analyzer by clicking on any <EXIT> button or by selecting the Exit option from the FILE menu item.

---

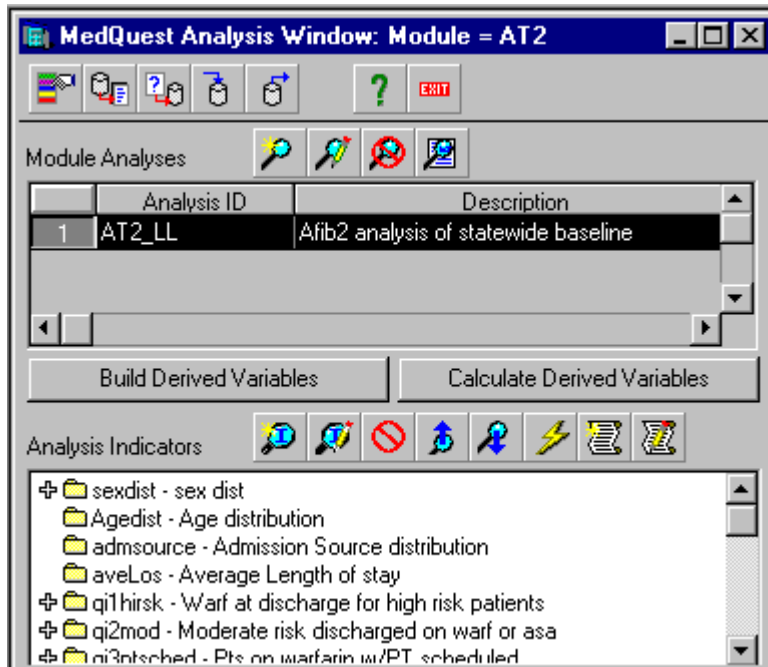
# DESIGN/CODE ANALYSIS

---

---

## ANALYSIS DESIGN WINDOW

---



When the **MedQuest Analysis Tool** screen is displayed, there are a number of functions that you can perform. These functions are all activated from icons or drop-down menus available on the screen:

**SELECT MODULE.** You can select another module to analyze by choosing this button.

**VIEW/PRINT REPORTS.** You can produce several reports from the dictionary. See the **Reports** section for a complete description of the reports that are currently available.

**QUERY VARIABLES.** This function enables you to browse through the variables and their clinical help for all of the modules listed in your project database. There are three search mechanisms you can use: 1) you can search for a variable by module when you know the screen and variable; 2) you can search for a variable using a word or words that search through all of the variables in a selected module; and 3) you can perform a search using the keywords that have been associated with the variable in a given module dictionary. You can add these keywords during the design to create a more robust dictionary variable classification scheme. See Appendix A for a complete discussion of the **Query Variables** function.

**IMPORT.** You can access the **Import** function through the MedQuest Analyzer. If you select this option, you will be taken directly into the MedQuest Manager's (SMS) **Import** function. This allows you to import data into the datastore database. See the MedQuest Manager's (SMS) Reference manual for a complete discussion of the **Import** function.

**EXPORT.** You can access the **Export** function through the MedQuest Analyzer. If you select this option, you will be taken directly into the MedQuest Manager's (SMS) **Export** function. See the Perform Analysis section for an overview of this feature. See the MedQuest Manager's (SMS) Reference manual for a complete discussion of the **Export** function.

**HELP.** You can access the MedQuest Analyzer application help from the icon and the other help from the drop-down menu. The General Help provides general MedQuest Analyzer application help. If you have compiled help from your dictionary (in MedQuest), you can view that help file (XXXDICT.HLP where XXX is the module acronym) through Module help. About (MedQuest Analyzer ) identifies software and dictionary version information.

**WINDOW DROP-DOWN MENU ITEMS.** You have the capability in analysis to open up to 10 views/graphs at one time. The options available in the **Window** menu include (basic Windows functions): Cascade; Tile Horizontal; and Tile Vertical. These options will arrange the views/graphs accordingly. This menu will also display the name of the module loaded into the MedQuest Analyzer as well as the names of the views/graphs that are open.

---

## ANALYSES

---

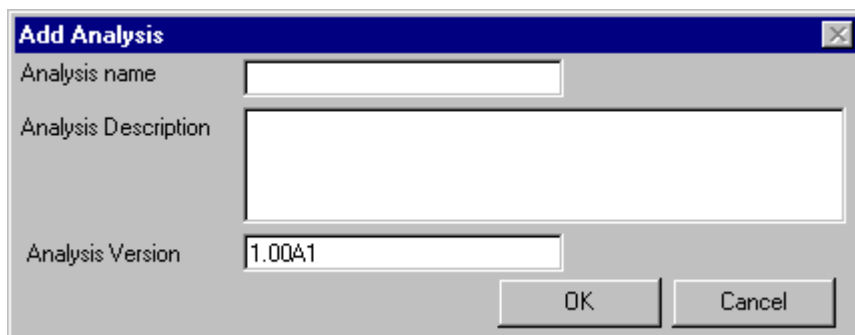
The MedQuest Analyzer is used to create sets of logic that utilize the clinical data to perform analyses (an analysis is a group of related items used to answer questions about the clinical data).

---

### Add an Analysis

---

Click on the <ADD ANALYSIS> button located next to the MODULE ANALYSIS menu bar to display the **Add Analysis** screen.



Enter a descriptive name for the analysis, an analysis description, and a version number (if different than the default).

**NOTE:**

You must add an analysis before you can add an indicator.

---

### Edit the Analysis

---

Clicking on the <EDIT ANALYSIS> button will let you change the analysis name, description, and version.

---

### Delete the Analysis

---

Select the <DELETE ANALYSIS> button to delete an analysis. All of the indicators, views and graphs associated with the analysis will also be deleted.



## Comment on the Analysis

---

Click on the <ANALYSIS NOTES> button to display the **Edit Analysis Notes** screen. These are any general notes you are interested in to describe the analysis.

---

## BUILD DERIVED VARIABLES

---

In performing an analysis, you may want to use the raw data to infer a given result, e.g., if a given clinical event such as an MI occurred. The MedQuest Analysis Tool enables you to do this by creating a derived variable. You can create a derived variable by building an equation that describes the result, e.g., you can build a derived rule that says "If ... and ... and ... then MI=true."

Performing an arithmetic or logical operation on another variable to derive a value creates a derived variable. Age and length of hospital stay are examples of derived variables: age is derived by calculating the difference between the date of birth and the date of some event, often the admission date. Calculating the difference between the admission and discharge dates derives length of stay.

Categorical variables are another type of derived variable. These can be created by grouping values into categories, e.g., categories for age may be grouped as follows: 1= < 65, 2= 65-74, 3= 75-84, etc.

In the analysis process, this step provides the basis for performing more complex queries by breaking the parts of the complex queries into smaller pieces. An analysis may often contain more derived variables than queries.

When you select the <BUILD DERIVED VARIABLES> button from the **MedQuest Analysis Tool** screen, the **Edit Derived Variables** window (as shown above) is displayed. The components are as follows:

**MENU BAR.** Displays the list of functions you can perform on the **Edit Derived Variables** window. Each of these functions contains additional options. You can display the available options using the mouse.

**DESIGN BUTTON BAR.** Displays the different functions you can use to design your derived variable. The buttons on the bar perform the same functions as the options listed under the DESIGN menu item. As you move your mouse across the button bar, a small box describing each function appears.

There are two types of design action buttons: direct action and action mode. The direct action buttons include: <ADD DERIVED VARIABLES>, <ADD DERIVED GRID VARIABLES>, <MOVE DERIVED VARIABLE UP>, <MOVE DERIVED VARIABLE DOWN>, <DELETE ALL DERIVED VARIABLES>, and <ADD/QUERY DERIVED VARIABLES>. The direct action buttons enable you to perform the function that is described on the button.

The action mode buttons turn on the selected action mode. These buttons do not directly perform the function; you must click on a variable to perform the function. However, these buttons allow you to continue performing the selected function without having to reselect the function button each time you switch to a different derived variable. The action mode buttons include: <EDIT DERIVED VARIABLE MODE>, <DELETE DERIVED VARIABLE MODE>, <EDIT DERIVED EQUATION MODE>, and <VIEW DERIVED DETAILS MODE>. After selecting these action mode buttons, you must select a derived variable in the *Derived Variables (single/grid)* list boxes on which to perform the function.

**DERIVED VARIABLES (SINGLE/GRID) LIST BOXES.** Displays the derived variables, if any, that are associated with each list box. In each list box you can perform a variety of derived variable design activities, depending on the current mode in which you are working. If you are using the **Edit**, **Delete**, or

**View Details** mode, then clicking on a derived variable will select that derived variable for the given mode's action.

## Add a Single Derived Variable

---

Click on the <ADD DERIVED VARIABLE> button to display the **Add Single Variable** screen. The screen that is loaded is the standard **MedQuest Variable Add/Edit** screen that you use for creating variables for a **MedQuest Data Entry System (DES)**. You should refer to Appendix A for more detailed instructions on creating and completing the variable parameters.

### Examples of Single Derived Variables

---

**Example 1.** After creating a derived variable for Age in years as (admission date - date of birth)/365.25, create a derived variable AGECAT that collapses into several categories. Create an option pick one derived variable with the following categories: 1= < 65, 2= 65-74, 3= 75-84, 4= >= 85, 0= UTD.

**Example 2.** Use the MedQuest discharge disposition categories (CDISDISP) and claim date of death to create a derived variable "Died within 30 days" (DIED30) where 0= still alive or date of death > 30 days from discharge, 1= died within 30 days of discharge.

**Example 3.** Create a numeric derived variable which assigns a value of 50 if the value of the hematocrit variable (CHEMACRI) >= 50 and is not missing or UTD.

**Example 4.** Create a string derived variable having the value "ANONYMOUS" if the last name is filled in and <> "ANONYMOUS".

## Add a Derived Grid Variable

---

The original data entry grid variables (incident or medication variables) for a given module are displayed in the *Derived Variables (grids)* list box. To add a derived grid variable, select a grid variable from the *Derived Variables (grids)* list box. Click on the <ADD DERIVED GRID VARIABLE> button to display the **Add Single Variable** screen.

#### NOTE:

Derived variables, whether being added to single or grid tables, are single variables created to perform an arithmetic or logical operation to derive a value.

The process for adding a derived **grid** variable is the same as adding a **single** derived variable.

### Examples of Derived Grid Variables

---

**Example 1.** Medications grid: If the parent variable is Medications at Admission, create a option variable that can be used as a flag (1=True, 0=False) if a medication record is present having the code for Aspirin.

**Example 2.** Incident grid: Create an option variable that can be used as a flag (1=True, 0=False) if a surgery was performed having a diagnosis code with the first two digits = 36.

**Example 3.** Incident grid: Create a numeric variable to determine length of surgery in minutes by subtracting time in from time out for each incident.

## Edit Derived Variable Mode

---

The **Edit Derived Variable** function switches the edit mode on or off. When you are in **Derived Variable Edit** mode, you can continue to edit any derived variable in the **Design** window by clicking on it. You will be able to continue to edit the derived variable parameters in the same way as described in the **Add a Derived Variable** or **Add a Derived Grid Variable** sections.

### NOTE:

The advantage of being in a specific mode is that you can quickly select different derived variables on which to perform a particular function.

## Delete Derived Variable Mode

---

When you select the **Delete Derived Variable** mode on the **Edit Derived Variable** window, you can select the derived variable that you want to delete. You will be asked whether you want to delete the contents of the derived variable as well. This means that all of the specific information you have entered to define that derived variable will be deleted from the dictionary and you will not be able to recover it. When a derived variable's contents are deleted, the derived equations associated with that derived variable will also be deleted. You do not have to accept this permanent deletion. When you are finished with the deletion, it will not appear in the *Derived Variables (single)/(grids)* list boxes.

## Edit Derived Equation Mode

---

Once you have created a derived variable of a given type, e.g., a number or option pick one, you can create the logic that will be used to derive the result of that variable by selecting the **Edit Derived Equation Mode** function. In this mode, you will be able to select a variable and, from the menu that pops up, select the option to add or clear an equation.

There are two types of derived equations you can add. The first type "Save ? to ME (Always)" lets you build an equation that calculates the value saved to the derived variable. For example, you might want to create a derived variable that stores the date/time of the admission date plus 48 hours. You would create this variable using this type of equation since the answer is always calculated the same way.

When you select this option, you will be presented with an equation builder (same as used in building rules in MedQuest) that enables you to build equations for the following variable types:

- Number (no units): A complex equation using numeric variable(s) and constants
- Number (w/units): A complex equation using numeric variable(s) and constants
- Date: A date variable plus/minus days
- Time: A time variable plus/minus minutes
- Date/Time: A date/time variable plus/minus hours

In addition to building the variable types listed above, you can also add the following derived functions to a derived variable that is type Number (no units):

- Age (from a date)
- Month (from a date)

- Year (from a date)
- Days between (two dates)
- Months between (two dates)
- Years between (two dates)
- Hours between (two date/times)

These "derived functions" are useful for deriving commonly used data elements from the raw clinical data. They are available in the equation editor by selecting the **Function** command.

For other variable types or for more complex logic, you may want to use the "Save ? to ME IF..." derived equation rule. This rule type lets you build a logical expression that will determine if a given value is saved to the derived variable. For example, you might build a complex rule that says "Save '1' (Yes) to 'Has an MI' IF variable1=True or variable2=True or (variable3=True and variable4=True)."

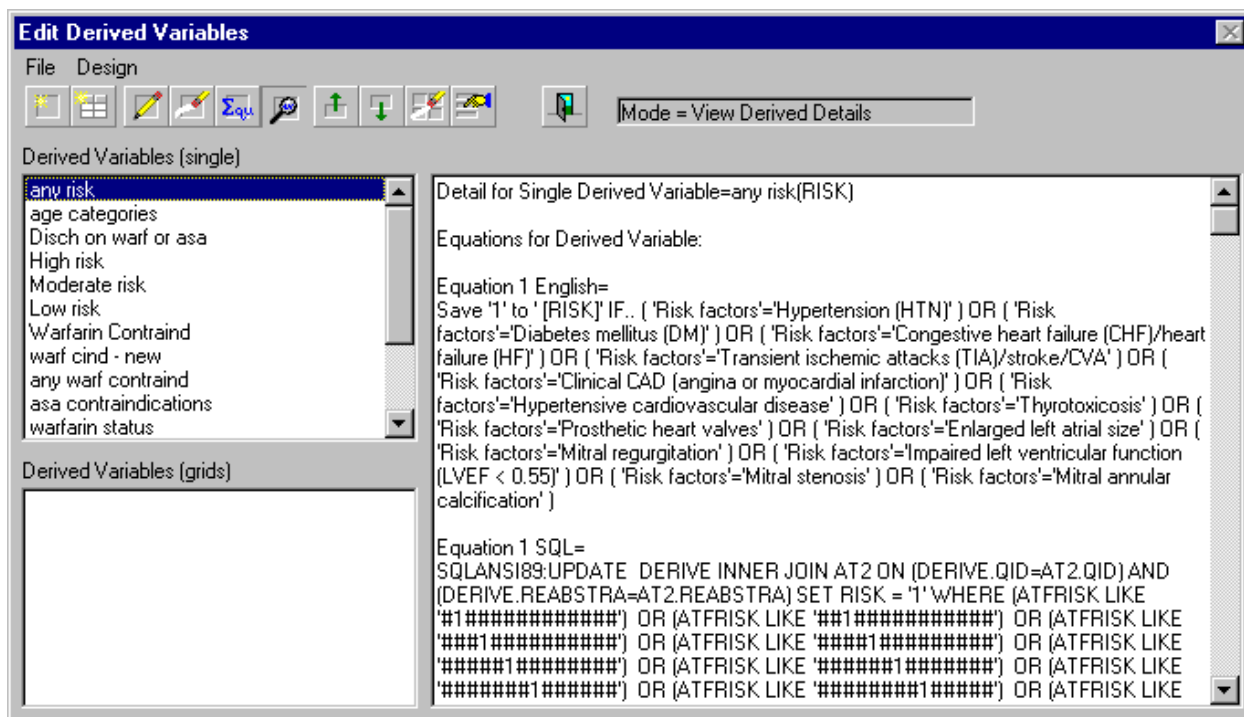
When you select this type of derived equation, you are presented with the same rule builder that you are provided in MedQuest. This rule builder lets you build one or more equations with up to 20 logical expressions. You can build any logical expression by selecting for each cell the variable you want, the operator you want and the value, variable or value plus variable that you want included in the expression. You can also use the operators (=, <>, <, <=, >, >=, AND, OR) and parentheses to build a more complex logical expression that will be used to derive the result.

When the result of your rule is true, the value that you want to be saved to the derived variable will be saved. Otherwise, the save action will not take place. The most common use of this type of equation is to establish whether a given set of logic is/is not true. In this case, it is often useful to just use one equation to save the value you establish as true to the variable. You may want to also have a second equation that saves the value you establish as false to the variable (the opposite of true) since there is no "Else Save" part to this rule type; however, this is often not necessary since later in the analysis you can easily infer that it is false if it is not in fact true.

## View Derived Details Mode

---

To view the details associated with a derived variable, click on the <VIEW DERIVED DETAILS> button and highlight the derived variable you want to view.



## Move Derived Variable Up

To move a derived variable up, highlight the derived variable you want to move up and click on the <MOVE DERIVED VARIABLE UP> button. This is useful if you want the derived variables to fire in a given order because you have one derived variable dependent on another.

## Move Derived Variable Down

To move a derived variable down, highlight the derived variable you want to move down and click on the <MOVE DERIVED VARIABLE DOWN> button. This is useful if you want the derived variables to fire in a given order because you have one derived variable dependent on another.

## Delete All Derived Variables

To delete all of the derived variables at once, click on the <DELETE ALL DERIVED VARIABLES> button. You will be asked to verify your deletion. The derived variables will not appear in the *Derived variables (single/grid)* list boxes on the screen. The **Delete All Derived Variables** function operates in the same manner as the **Delete (Single) Derived Variable** mode.

In addition, you will be prompted to choose whether you want to remove the contents of the derived variables from the dictionary. If you choose "No," then the variables you deleted will remain in the dictionary for reuse at a later time. If you choose to delete the contents of the variables, all of the analysis equations associated with those variables will also be deleted.

## Query/Add Derived Variables

---

To query/add derived variables, click on the <QUERY/ADD DERIVED VARIABLES> button to display the **Query/Add Variables** screen.

You can query for variables to retrieve in other modules and add them to the **Edit Derived Variables** window. This function can also be used just to query other variables in other modules (e.g., to quickly view the help for a variable) and to retrieve (add) help for another variable to the workspace where you are editing help. The objective of the general query interface is to provide you common access to variables in other modules.

For more information on using the **Query/Add Variables** function refer to Appendix A in this manual.

## Exit Build a Derived Variable

---

To exit the **Edit Derived Variable** screen, click on the <CLOSE> button or click on the FILE menu item and select the Close option.

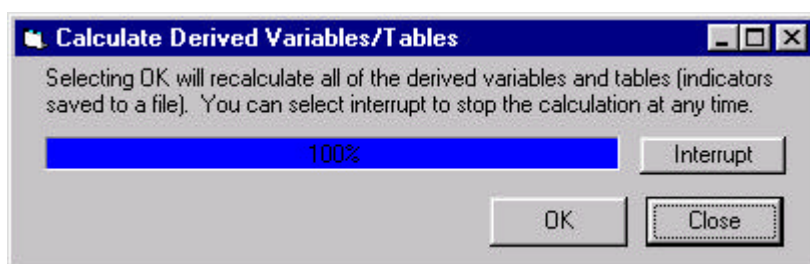
---

## CALCULATE DERIVED VARIABLES

---

You may calculate the derived variables once a derived variable has been created. You must recalculate the derived variables whenever a new derived variable is added, therefore, it is advantageous to create as many derived variables as possible at one time.

To calculate the derived variables, click on the <CALCULATE DERIVED VARIABLES> button to display the **Calculate Derived Variables/Tables** popup box. Click on the <OK> button to begin the calculation. You can stop the calculation at any time by clicking on the <INTERRUPT> button. Once the calculation has been completed, click on the <CLOSE> button to return to the **MedQuest Analysis Tool** screen.



### NOTE:

If clinical data is added after you have started the analysis, you should rerun the calculation of the derived variables.

---

## INDICATORS

---

In the MedQuest Analyzer, you build your query of the data by creating an **Indicator**. When you add (or edit) an indicator, you are selecting the variables that you want in your query. The analysis tool will generate an SQL statement for you after you have completed the indicator (query) building template. To build a query you are given the options to describe (in equation form) the records you want to include or exclude, the variables you want to display, the variables you want to group by (and filter for), and the

variables by which you want to order by. In addition, you are given the ability to identify variables that you want to include in dynamic queries during the analysis, e.g., allow the user to vary the diagnosis code that is included (or excluded) in the analysis.

The results of the indicators will be displayed in a data table and can also be displayed graphically.

An indicator can be used to measure a process or an outcome associated with the effectiveness of an intervention. It is a logical statement that can evaluate to true or false, or can be broken into categories. In order to determine the measure, two or more populations may be compared. For example, clinical indicators for the Cooperative Cardiovascular Project (CCP) can be used to measure the reduction in mortality resulting from the use and timing of aspirin and thrombolytic treatments in the acute phase of a myocardial infarction. By comparing a hospital sample to a sample of records collected at the state or national level, the analyst can determine the effectiveness of the treatment. Other indicator results obtained from the same data include reduced mortality from the use of beta blockers as well as from the long term use of aspirin.

## Add/Edit an Indicator

---

Click on the <ADD INDICATOR> button to display the **Add Indicator** screen. You must add an indicator before you can add a view, graph, or script.

When you add or edit an indicator, the primary query building screen will appear. On this "query template" building screen you will be asked to enter the pertinent information to create an SQL query for the data you want to analyze. Following is a detailed breakdown of the types of information you are required to fill out on the template.

You must enter a unique name and provide a useful description for each indicator you create. You may also want to modify the version number for version control purposes.

To create an indicator, you must have at least one variable for which you can specify logic, view, group by, or order by.

## Save Indicator as Derived Table

---

When you select this option, the indicator you build will be saved as a table in the data store database using the indicator name as the table name. This option can be useful for building a "data warehouse" table on which further analysis can be performed. One advantage of this table is that, for larger datasets, the performance speed of loading the indicator will be improved. Another advantage is that you can create one indicator table and then create another indicator based on that table, and continue to subset your analysis in this manner.

There is one limitation in the use of a derived table. Once you create an indicator as a derived table, that table cannot be linked or joined with other data in the data store database. This results in an isolated set of data for analysis.

## Indicator Type

---

The user has two types of indicators to choose from, **General** (general SQL query) and **Percentage Breakdown**. The **Percentage Breakdown** indicator produces a report of the counts for each combination of variables selected on which to report.



## ***Records to Include/Exclude***

---

When you start building the logic for an indicator, the first thing you want to determine is which records you want included or excluded in the set you are analyzing. The standard MedQuest rule editor is used to build this logic. You can select any variable that is in the dictionary, e.g., you might build a rule that says to include all records where the Systolic BP < 100 and the "Patient had Internal Bleeding" is true.

You can also build very complex queries using this rule editor, e.g., you can build a rule where you ask for all records where an admission medication was given within 48 hours of a discharge medication. In this type of query, the MedQuest analysis software will automatically create a complex table join and all you have to do is establish the logic you want to execute. It is important to point out the risks associated with this flexibility. If you use a large number of grid type variables and try to include variables from too many grids in building an include/exclude rule, then the tables that are joined together may result in either too much data (linking multiple tables to multiple tables) or data results that are correct but intuitively do not make sense; therefore, be careful to only include grid variables in a rule that logically belong together.

## ***Variables to Display***

---

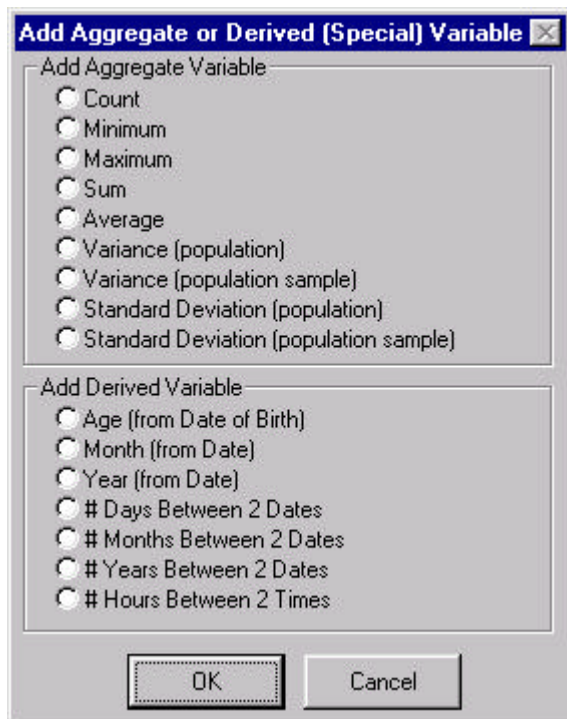
After identifying the records to include/exclude, the next step in building a query is to identify the variables you want to display during analysis. In the analysis tool, you can add (+) or delete (-) variables to view or build a set of variables or add a special type of variable.

When you build a set of variables, you are presented with a screen that lets you select all of the variables from a given screen, screen set, or grid that are in the dictionary. This allows you to select your data as it is functionally organized for data entry. In some cases, this approach is useful when the data entry screen design matches the way the data elements are organized for analysis.

You can also select a set of variables from a specific table that is currently available in the data store database, e.g., you may want to include all of the variables from a table "FACILITY" that you added to the data store database. This "other" table and the variables in it can be accessed in this way.

When you add a special variable, you will have the option to add an aggregate or derived variable.





When you build one of these variable types, you will be prompted to select a variable (of the correct required type) that you want to use, e.g., to derive the number of days between dates you will be prompted to enter two dates.

You also have the ability in selecting the variables to identify if you want to select just the unique (for the variables listed) set of the variables or the unique variables where the entire row is unique. This is useful in analysis for identifying the different unique values that have been entered for a given variable.

## Percent Breakdown

---

If the Percent Breakdown indicator is selected to report by, the user would enter the variables on which to report. For example, the user might want a breakdown by gender of the number of patients with MIs on admission.

## Group Records By

---

When you are performing an analysis, it may be necessary to organize your data by different units such as by state, by month, or by year. When you group variables, you have the ability to add a variable (+) or delete a variable (-) and to add special variables (as described above). When you group records, you are building an "aggregation" of the data and rules may be applied to other variables included in the indicator, e.g., grouping by state and then including admission date in the variables to view; the admission dates cannot be grouped by state (unless they are included as one of the variables to group by along with state).

You can include additional logic that refines the records you want to group by ("Having"). When you select this logic you will be presented with the standard MedQuest equation editor and the ONLY variables you will be allowed to select from are those variables that you are grouping by.

## ***Order Records By***

---

You can select the order in which the data is presented in the indicator by selecting the variables to order by.

## ***Variables to Include in Dynamic Queries***

---

When you perform analysis using an indicator, you may find it useful to change the results slightly to determine the impact of different variable results on the table. You can establish the variables that you want to change by adding variables to *The Variables to Include In Dynamic Queries* list box. During the analysis, you will be entering values for the variables that are in the list and building a dynamic include/exclude option for the logic.

## ***Manually Edit SQL***

---

If you can program in SQL and want to build your own SQL statement (e.g., include a Visual Basic function that is not code generated using the indicator builder), you can select to edit the SQL. Note that manual editing of the SQL in terms of the organization of that SQL and its pertinence to the indicator concept you are trying to represent is not directly supported, e.g., you will not be able to translate the SQL statement you build into an English version of the indicator.

## ***Other Functions***

---

You have the ability to preview the SQL, preview an English language version of the SQL (if you have not manually edited the SQL) and save your indicator logic.

When you are finished, you will have an SQL statement that will be used during analysis to return the results from the current data state. This is the primary purpose of the MedQuest analysis tool: to help you quickly build an SQL statement that represents the logic in your indicator. If you cannot represent your indicator in a single SQL statement, then you may want to move to using other software analysis tools.

## **Delete Indicator/Graph/View/Script**

---

To delete an indicator, graph, view, or script, highlight the desired indicator, graph, view, or script you want to delete and click on the <DELETE INDICATOR/GRAPH/VIEW/SCRIPT> button.

## **Move Indicator/Graph/View/Script Up**

---

To move an indicator, graph, view, or script up, highlight the desired indicator, graph, view, or script you want to move up and click on the <MOVE INDICATOR/GRAPH/VIEW/SCRIPT UP> button.

## **Move Indicator/Graph/View/Script Down**

---

To move an indicator, graph, view, or script down, highlight the desired indicator, graph, view, or script you want to move down and click on the <MOVE INDICATOR/GRAPH/VIEW/ SCRIPT DOWN> button.

---

# PERFORM ANALYSIS

---

---

## ANALYZE

---

Any object may be analyzed by double clicking on that object or by highlighting the desired indicator, graph, or view to analyze and clicking on the <ANALYZE> button. The **Analysis (Indicator/View/Graph)** window will pop up after a few seconds. At this stage you can view the data in table and/or graph format, and also dynamically modify the logic to change the results.

---

## VIEW TABLE RESULTS

---

For viewing the query results, the data will be displayed in spreadsheet form as shown below.

---

## Add/Edit View Logic

---

In performing an analysis, you may decide to make changes to your query of the data to observe the differences in the results. In the MedQuest analysis tool you can accomplish this using the **Edit View Logic** feature. When you edit the view logic, you are simply adding additional logic to your indicator query that further refines the data for you.

Each time you change the logic, the table that you view during analysis will change (and if you have a graph currently defined, the graph results will change also). You can continue to make these logic changes until you find a result that you want to save. When you want to save your logic change, you can save it either as a VIEW or as a GRAPH (if you have a graph specified). What occurs in the program when you save a VIEW or GRAPH is the logic (query) for the current SQL statement is saved in the dictionary for that object.

This option provides a lot of flexibility in establishing a set of VIEWS and GRAPHS that will be associated with the indicator. The analysis strategy is to organize the views and graphs in a logical way for presentation to other researchers.

When you modify the indicator logic, you are only allowed to make three types of modifications to the indicator logic (this constraint insures that you create a new indicator for a major modification of the data query). The three options appear in the **Edit View Logic** design window and are described below.

---

## Build Dynamic Query

---

If you add variables to the dynamic query during the creation of the indicator, then those variables will be presented as you edit the view logic. You can then build a logical equation that further refines the records to be included/excluded from the data query. To build the query, select an operator (=, <>, <, <=, >, >=), a value for the variable, and a logical expression (AND/OR, if necessary for linking expressions together). A value for each variable specified is not required. You must have an AND/OR linking variables+operators+values if you have more than one expression in the query.

To demonstrate, below is an example of a dynamic query that can be built:

- Admission diagnosis = 333.33 And
- Systolic BP =
- Discharge diagnosis <> 333.33
- Discharge Systolic BP =

This logic will include records where the **Admission Diagnosis** is 333.33 and the **Discharge Diagnosis** is not equal to 333.33 and will ignore the other variables since they are not included with values. This flexibility enables you to quickly modify the variables included in the set added to the indicator. The variables specified in the query should be those that are pertinent to your analysis.

## ***Change Logic for Records to Include/Exclude***

---

The dynamic query changes described previously are limited to accepting a single value for each variable and are designed to provide for quick adjustments in the logic. If you want to build a more complex equation to specify what records to include/exclude, you can change the logic for the records to include/exclude. This option presents you with the standard MedQuest rule editor and enables you to modify the current logic for the indicator, view or graph on which you are working.

Note that you can use parentheses and AND/OR expressions between the variables to build a more complex set of logic. You can also specify a more complex set or range of diagnosis codes.

The flexibility provided in this option will enable you to build a complex include/exclude expression that changes the original concept you may have had for the indicator. For example, the original indicator may not have included any logic for **Admission Medications** (a grid) and you might use this feature to add such logic. Since the admission medications are in a grid, they will be joined with the other data and result in a data table that is driven by the number of medications you have in the grid. This result is the data linking "multiples" in the analysis. This activity can be easily accomplished using this tool; however, if this was not the original intent of the analysis, it may be advisable to back up and change the indicator logic (since you have made a fundamental change) or create a new indicator that reflects your new concept.

## ***Change Variables to Order By***

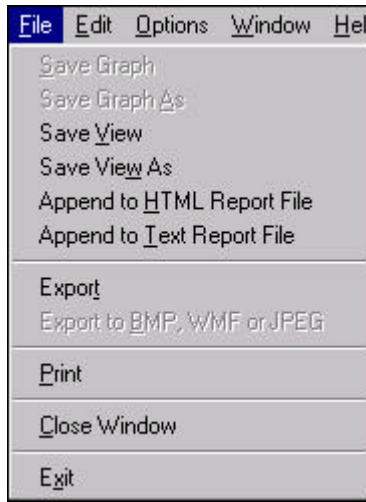
---

You are given the capability to select other variables specified in the logic to order by.

The logic adjustment options described above provide you with the flexibility to use one or more of them in various ways, e.g., you may find it easier to use the dynamic query option all the time and never use the ability to modify the records to include/exclude option. You can determine the approach that works best for you.

## Save View

---



After viewing the table results of your query, you may save it by selecting the Save View or Save View As options from the FILE menu item. Enter a name for the View in the text box and click on the <OK> button to save your view or the <CANCEL> button to discard the view.

This will save the current table view you are displaying for the indicator.

## Comment on the View

---

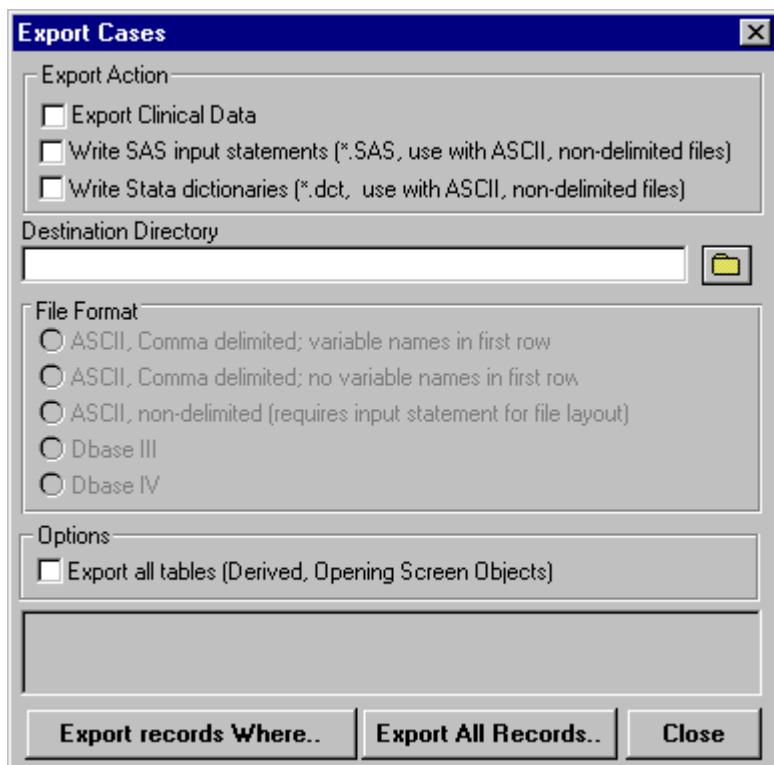
After a view table has been created, you can add notes to the view by clicking on the <NOTES> button to display the **Edit Indicator Notes** screen. After entering your comments, click on the <OK> button to save your comments and return to the view of the table.

## Export

---

To export the data in the currently displayed table or graph, select the Export option from the FILE menu item: Select one or more of the export actions, choose a destination directory, and then, if necessary, select the file format of the exported data. The various options for file formats are: ASCII comma delimited with or without variable names in the first row, non-delimited ASCII, dBase III and dBase IV. dBase III and dBase IV formats are general, easy to use formats that are supported by a variety of popular spreadsheet, word processing and database programs. When you click on the <OK> button, the export will begin. If you are exporting an indicator, view, or graph, one file for each export action will be created. If you are exporting from the MedQuest Analysis Tool window, all tables in the module dictionary will be exported.

Select the Export function and follow the instructions below.



## ***Export Action***

---

Check one or more of the following options:

- **Export Clinical Data.** Checking this option enables the File Format options. This is used for the most basic of ASCII or dBase exports.
- **Write SAS Input Statements.** When this option is selected, the MedQuest Manager (SMS) will generate SAS input statements in the list input format based on the natural order of the variables. It will also calculate the length of each record for the input statement. All the fields, including dates, are assigned the \$CHARn format.
- **Write Stata Input Statements.** This option works in a manner similar to the above SAS input statements. However, the variables stored as Memo fields are excluded from the input statement, as are user-defined labels that are excluded on the data entry screens in MedQuest.

## ***Destination Directory***

---

Type in the name of the directory where the files are to be stored or click on the folder icon to select a directory as the destination directory.

## ***File Formats***

---

There are several formats in which the exported data can be saved. Together, they will be placed in the destination directory. These file formats are available when the Export Clinical Data option is selected.

- **ASCII, Comma Delimited; variable names in first row.** Exported files are assigned the name of the table and the extension ".CDF". For example, a module XXX might export data to the following

files: XXX.CDF, XXXS1.CDF, XXXML1.CDF and XXXML2.CDF. String type variables are enclosed in quotation marks to handle any commas that may appear within the data. The first line of each CDF export file will contain the fieldnames of all the variables and will correspond to each data element in the export file. This file format is recommended for creating an import file for the MedQuest Manager (SMS).

- **ASCII, Comma Delimited; no variable names.** With the exception of not including variable names in the first row, this export is the same as the comma delimited export described above. This file format also has the ".CDF" extension.
- **ASCII, non-delimited (requires input statement for file layout).** This is useful for reading data into analytic software with a record layout (such as SAS or Stata) using the SAS input statements or Stata dictionaries described above. This file is in fixed-width format and has the ".TXT" extension.
- **dBase III/dBase IV.** Tables are exported to XXX.DBF, XXXS1.DBF, XXXML1.DBF, etc. for both versions of dBase. Memo fields are exported as well.

**NOTE:**

Selecting either of the dBase file formats will automatically default to export all tables including derived and Opening screen objects and the user will only have the option of exporting all of the records.

## Options

---

The user also has the option of including the tables that are created for derived variables when using the **MedQuest Analyzer** and the **Opening** screen objects developed in MedQuest to export. See the Analysis Reference Manual for complete details about derived variables and the MedQuest Reference Manual for complete details about the **Opening** screen objects.

## Export

---

Clicking on the <EXPORT ALL RECORDS> button will export all of the records in the specified file format. Clicking on the <EXPORT RECORDS WHERE> button will take the user into the **Edit Rules** screen. From here the user can define the cases to export based on a set of logic. For example, to export cases determined by a specific date, the user could enter a rule to export cases equal to a certain Discharge Date. This is a dynamic query create to export a record(s) once. After the export process, the rule/query is not saved.

**NOTE:**

The **Edit Rules** screen is the same Rule Editor used in the MedQuest software. Refer to the MedQuest Reference Manual for complete instructions on how to use the Rule Editor.

After completing a rule for exporting specific records, a popup box asking "Query will result in X records to export. Export?" will be displayed. Click on the <YES> button to complete the export or the <NO> button to cancel the export. When the export has been completed, a popup message box will state that the export process has been completed and it will list the files that were created. Click on the <OK> button after viewing the message.

To exit the **Export** program function, click on the <CLOSE> button.

## *Name of Exported Files*

---

There are two different types of exports that may be performed from the Analysis: the export of a whole module's data; and the export of a single indicator, view or graph. When exporting the whole module, the filenames will consist of the name of the table from which the file was created followed by the extension, depending on the file format chosen to export to. When exporting an indicator, view, or graph, the filename will be that of the module followed by "\_SQL" and then the extension. These three types of data are all handled in the same manner.

### **EXAMPLE:**

Assuming a module has tables named CCP, CCPML1, CCPML2, DERIVEML1 and DERIVEML2, the files that are created for a comma-delimited export when exporting the whole module will be CCP.CDF, CCPML1.CDF, CCPML2.CDF, DERIVEML.CDF and DERIVEM2.CDF. Please note that for table names that are longer than eight characters, the tables will be truncated to allow for a number to be appended to the filename to prevent one file from overwriting another. This is why the DERIVEML1 table is saved as DERIVEML.CDF and DERIVEML2 table is saved as DERIVEM2.CDF.

### **EXAMPLE:**

When exporting an indicator, view, or graph from the CCP module, the file that is created will be CCP\_SQL.CDF.

## **Print View**

---

To print the currently displayed view/table, click on the FILE menu item and select the Print option.

## **Close Window**

---

To close the table view, double click on the upper right or left hand corners of the **Tables** screen (Windows function) or click on the FILE menu item and select the Close Window option.

## **Exit**

---

To exit the MedQuest Analyzer, click on the FILE menu item and select the Exit option.

## **VIEW GRAPH RESULTS**

---

When you are working with an INDICATOR or a VIEW in analysis, you may want to present the data in a graphical format. The MedQuest Analyzer enables you to create any number of graphs for both the indicator logic or view logic. You are allowed to make given basic variations in the graph (such as the graph type and the data to include in the graph) and save the graph.

You can load a graph for an analysis by double clicking on the graph object. You can also add a graph to an indicator or view by double clicking on either of those objects and then selecting the graph tab.

To display a graph beneath an indicator or a view, click on the plus sign (+) to the left of the indicator or view. If there is no plus sign (+), there are no graphs to display.



## Data to Graph

---

To begin creating a graph, click on the <DATA TO GRAPH> button and follow the steps below.

### Variables Selection

---

- Select the "Graph" tab displayed behind the results table. The graph tab will move in front of the table, and a blank area will be displayed.
- Click on the <DATA TO GRAPH> button. A set of tabs marked "Variables" and "Retrieve" will appear. The tab marked "Variables" will be on top.
- Select one or more numeric fields to be plotted along the Y axis from the box labeled "Fields Available". Highlight the items in the list to be selected, then click on ">>" to select the item; "<<" to unselect the item.
- Select the numeric field (if any) to be plotted along the X axis. Click on the downward arrow on the right of the area labeled "Field for X axis", then click on a fieldname. This is necessary only when you are plotting X versus Y data.
- Select either the same field or a different field to display as X axis labels. This field can be numeric or alphanumeric. If you select "NONE" the X axis will be labeled with a counter; in this case, the software will determine how the X axis should be displayed.

## Graph Settings

---

When you click on <GRAPH SETTINGS> a set of tabs is displayed. These tabs contain options that enable you to customize the appearance of the graph. A context sensitive help file is included with the graph. When you click <HELP>, a picture of the property window is displayed. By using the mouse to click on an object or tab in the property window, the help for that object will be displayed.

### NOTE:

Not all of the options that appear in the help file for the graph are available at this time.

From the **MedQuest Analysis Tool** screen, select an indicator. It will take a few seconds for the system to execute the SQL query and produce a table with the results.

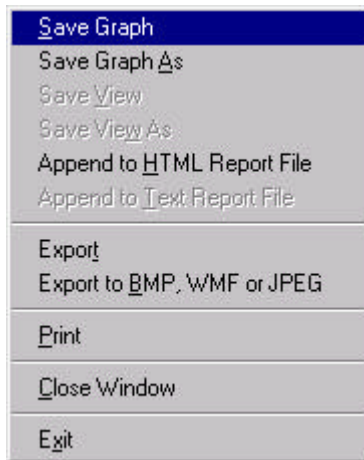
## Retrieve a Graph

---

Click the tab marked "RETRIEVE". The list of graphs that have been created and stored for that indicator will be displayed. Select a graph to view, click on the <RETRIEVE SETTINGS> button, and then click on the <OK> button to view the graph.

## Save a Graph

---



After viewing the graph results of your query, you may save the settings by selecting the Save Graph or Save Graph As options from the FILE menu item. Enter a name for the graph in the text box and click on the <OK> button to save your graph or the <CANCEL> button to discard the graph. This will save the current graph you are displaying for the indicator.

## Comment on the Graph

---

After a graph has been created, you can add notes to the graph by clicking on the <NOTES> button to display the **Edit Graph Notes** screen. After entering your comments, click on the <OK> button to save your comments and return to the view of the graph.

## Export to BMP, WMF, or JPEG

---

This feature will save the currently displayed graphic image to a .BMP file, a .WMF file, or a .JPEG file. BMP files (pictures) and WMF files can be brought into a word processing document as well as a graphics package (that supports BMP and/or WMF file structures) for editing. The JPEG export format is used to save the graph if it is to be included in an HTML document.

## Print Graph

---

To print the currently displayed graph, click on the FILE menu item and select the Print option.

## Close Window

---

To close the graph view, double click on the upper right or left hand corners of the **Graphs** screen (Windows function) or click on the FILE menu item and select the Close Window option.

## Exit

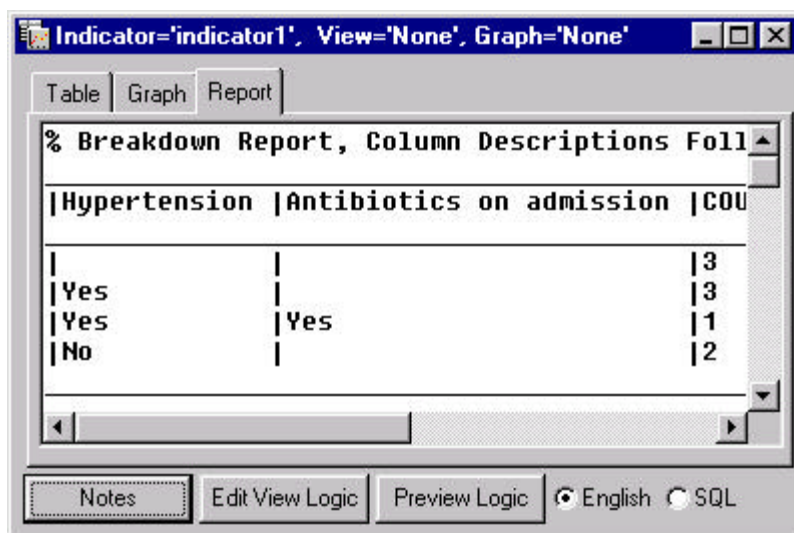
---

To exit the MedQuest Analyzer, click on the FILE menu item and select the Exit option.

---

## VIEW REPORT RESULTS

---



The user can also view the indicator in a report format. When the **Report** tab is selected, a textual representation of the data is produced. This report enhances the table view by looking up the descriptions for values. For example, the table view would display the response to a Yes/No question as 1 or 2, while the report view will display the response as "Yes" or "No."

### NOTE:

For a Percentage Breakdown indicator, the report view is the default view that will be displayed since the table view is less meaningful.

---

## Append to Text Report File

---

When the option to Append to Text Report File is selected, the user can append the currently selected report to a text file. The report can then be modified to fit the user's needs.

---

## BUILD/RUN ANALYSIS SCRIPT

---

In performing an analysis, you will be creating and modifying the three basic analysis objects: Indicator; View; and Graph. However, you may want to organize these different sets of objects into different sequences for presentation to explain the results of the analysis. This can be done by using the SCRIPT object. In this object type, you can add any indicator, view, or graph in the analysis to the script and then allow the user to "play back" the script in the sequence you constructed.

---

## Add Analysis Script

---

Click on the <ADD SCRIPT> button located on the ANALYSIS INDICATOR menu bar to display the **Add Script** screen.

Enter a name for the analysis script in the *Script Name* text box. This field can be up to 12 characters in length. Enter a description of the analysis script in the *Script Description* text box. You may change the

version number of the analysis or use the default version number that is displayed in the *Script Version* text box. The version identifier can be up to eight characters in length.

After completing the necessary information, you must select an object from the *Objects to Add to Script* list box and add it to the *Script Components* list box. An object is a view (table), a graph, or analysis notes. Using the >> and << buttons, you can move objects to and from the *Script Components* list box; however, an object cannot be added to the *Script Components* list box more than once. Using the <UP> and <DOWN> buttons, you can change the order of the objects in the *Script Components* list box. (You cannot change the order of the objects listed in the *Objects to Add to Script* list box.) When you have finished adding or changing the order of the components, click on the <OK> button to save your changes and close the window, or click on the <CANCEL> button to discard your analysis script and close the window.

## Edit Analysis Script

---

To edit an analysis script, click on the <EDIT SCRIPT> button to display the **Edit Script** screen. The **Edit Script** screen has the same look and functionality as the **Add Script** screen. Once you have completed all of your edits, click on the <OK> button to save your changes and close the window or the <CANCEL> button to discard your changes and close the window.

## Run Analysis Script

---



Running an analysis script is like viewing a slideshow of the elements that make up the script. However, the windows in which the script elements are displayed in are shown in the main **MedQuest Analyzer** window. The "remote-control" window to the right controls the running of the script. This window contains a list of the components in the script as well as buttons to manually control which component is displayed or to go through all of the components and display each one in order at a specified time interval.

Each component (graph or view table) is displayed in a new window and can be centered horizontally and vertically or displayed  $\frac{1}{4}$ ,  $\frac{1}{2}$  or  $\frac{3}{4}$  the size of the main **MedQuest Analyzer** window. These options are selected using the settings button on the lower right-hand side of the window.

The components are displayed, using the play button, in order from top to bottom. Using the forward, back, first and last buttons, you can manually move through the components and display them. Using the jump to button, you can select a component from the list and display it without having to cycle through and display all of the ones before or after it.

A graph or view table will be loaded once it has been displayed. If you display it again, and it is already loaded, it will be made the top window instead of being loaded again. This activity helps to save memory and resources while viewing scripts.

The following buttons are available for controlling the script before and while it's running.



Stops the current playback of the script and resets to the first component.



Makes the first component in the script the current one and loads it.



Stops the current playback and closes the window.



Makes the last component in the script the current one and loads it.



Plays the script from the current component.



Shows the previous component in the script.



Pauses the current playback.



Shows the next component in the script.



Edits the settings for the playback of the script.



Jumps to the selected component.

---

## CAPTURE ANALYSIS TO WEB PAGE (HTML)

---

Using the Append to HTML Report menu option, you can send the current graph or view table to an HTML file. When you choose this item from the menu, you are prompted to select an HTML file to save to. If the file already exists, it will be replaced. If you are saving a graph, you will then be prompted to enter the name of the JPEG graphic file to which the image of the graph will be saved. Once this is done, the HTML file is opened and can be appended to further.

While the HTML file is open, the above window is displayed to show that the file is open and being written to. When this window is closed, the HTML file is closed and saved to disk. Once a file has been closed, it **cannot** be appended to again. (This may change in a future release of the MedQuest Analyzer.)

---

## Append to Text Report File

---

When the option to Append to Text Report File is selected, the user can append the currently selected report to a text file. The report can then be modified to fit the user's needs.

---

# SENDING ANALYSIS TO SOMEONE ELSE

---

---

## SENDING A DICTIONARY/ANALYSIS

---

To send a module dictionary that also contains analyses, you should send the dictionary database (XXXDICT.MDB where XXX is the three-letter abbreviation of your module), e.g., if your module is named ABC then the dictionary database is ABCDICT.MDB. This file contains the Dictionary and any Analyses in that dictionary.

---

## SENDING DATA

---

The data for a module is contained in the clinical database or datastore named "XXX.MDB" where XXX is the three-letter abbreviation of your module, e.g., if the module is named ABC then the datastore database is ABC.MDB. It is most useful to the recipient of the data that he/she also has the dictionary (ABCDICT.MDB) for the module since understanding the data may require the information provided in the dictionary.

---

## SENDING ANALYSIS SCRIPTS

---

The analysis script definition is contained in the dictionary database XXXDICT.MDB (where XXX is the three-letter abbreviation of your module). However, the data in the analysis script are contained in the clinical database XXX.MDB (where XXX is the three-letter abbreviation of your module). Both of these files will be necessary to send the analysis scripts to another researcher.

In the future, it will be possible to save an analysis script to a web page (HTML) file. This functionality will be added at a later date.

---

## SENDING WEB PAGE (HTML) OF ANALYSIS

---

Sending a web page is slightly more difficult than performing the other activities of an analysis. A web page contains one .HTM file and may also have one or more graphic files containing screen shots of the graphs in the analysis. The MedQuest Analyzer saves these graphics files as JPEG images that have the .JPG extension. It is up to the designer/creator of the web page to keep track of which .JPG files belong to which .HTM files; otherwise, the .HTM file will need to be searched for all .JPG files that are listed in it. You may want to store each .HTM file in a different subdirectory to organize a web page and its associated .JPG files.

---

# REPORTS

---

---

## VIEW/PRINT REPORTS

---

The MedQuest Analysis Tool provides you with the capability to produce reports that may be used to facilitate the collaborative system design process and analysis of the data.

The clinical data dictionary contains all of the information necessary to use the data entry system that you have designed. At any point during the design process, you may decide to document the variables you have created to provide a record of the data elements and their attributes. The document generated from the data dictionary during development helps facilitate communication among team members, as well as allows you to keep track of changes at different points in the process. You can accomplish this by viewing reports of the individual variables or by comparing one version of the data dictionary to another version to identify the differences.

The data dictionary reports present the key elements of the data dictionary without requiring knowledge of the technical structure of the data dictionary. The report can display the elements of the data dictionary along with their attributes in a concise and readable standard format.

---

## SELECTING A REPORT

---

When you select the **Reports** function, you will be presented with a report selection window where you must select the type of report you want to produce.

A report can be produced in five ways:

- Rich Text Format (RTF) File, which is good for importing into a word processor;
- Text File;
- Report to the Printer;
- Report to the Screen; and
- HTML.

For each of these modes, a temporary file is always produced. Therefore, it is necessary to specify the location and the name of the file if you want to change the default file location and the filename provided by the MedQuest Analysis Tool.

If you decide to send your report to the printer, you also have available the option to setup the printer using the function provided.

After selecting the report, select the <OK> button to begin generating the report or, if applicable, go to the report specific selection criteria associated with the report you selected.

---

# DICTIONARY REPORTS

---

## Dictionary Variables Report

---

This report lists all the selected module variables sorted by the **Short title**. The report also includes the 8-character field name (**Fieldname**), the module acronym, and the type of variable. An asterisk will denote all mandatory variables (\*).

When you select this report, you have the option of sorting the variables by natural order by screen or alphabetically by short title.

## Dictionary Detailed Report

---

This report displays the detailed information that you entered into the dictionary database during the design process. When you select this report type, you are presented with a set of options from which to select:

**ALL SCREENS.** This option reports all of the variables on all of the screens.

**SELECT SCREEN SETS/SCREENS.** You can use this option to select specific screens on which to report by selecting the screen set and screens that are in that set. Note, you must select at least one screen if you choose this option.

**INCLUDE CLINICAL HELP.** This option reports all of the variables on the selected screens and includes their clinical help.

## Dictionary Comparison Report

---

Often in the design process, it may be necessary to compare a previous version of the dictionary with the version in which you are currently working. This report enables you to compare any two dictionaries of the same module. To obtain a meaningful result from the comparison, it is recommended that you create a new directory and store your point-in-time dictionary. When you are ready to compare versions, indicate the locations and the filenames of the current module and the previous module.

When you produce this report, you can select any or all of the tables associated with your dictionary for comparison. By having the option of selecting specific tables, you can immediately review the changes to a particular area in your dictionary, e.g., if you select the Equations table, any differences in the data entry rules will be listed in your comparison report.

The data dictionary tables include:

- Variables in data entry system (MASTER)
- Screens in data entry system (SCREENS)
- Sets of screens in data entry system (SCREENSETS)
- Variables on screen(s) (VIEWS)
- Variables on grids (MULTVIEWS)



- System data (SYSTEM)
- Data entry rules (EQUATIONS)
- Classification types (CLASSES)
- Variable classifications (FIELDCLASSES)
- Numeric conversion groups (tblConvertGroups)
- Numeric conversion units (tblConvertUnits)
- Numeric conversion factors (tblConvertFactors)

---

## Clinical Help Comparison Report

---

This report compares the clinical help for a previous version of the dictionary with the version in which you are currently working. To obtain a meaningful result from the comparison, it is recommended that you create a new directory and store your point-in-time dictionary. When you are ready to compare versions, indicate the locations and the filenames of the current module and the previous module.

This report displays the variables whose help is different between the two versions.

---

## Data Analysis Variable Report

---

This report lists all of the variables for each table in the selected module in their natural order for which data can be entered. It also lists attributes of the variables that are required for performing analysis. These attributes include: field length, data type, format, and definitions for categorical variables. Variables that cannot easily be input into statistical software, i.e., "Memo" fields, are not described but are denoted by "\*\*\*".

---

## Data Entry Rules Report

---

This report lists the English translation for any of the selected criteria: all variables' rules, one variable's rule(s), or a selected rule type. When all rules are selected, they are listed in the order in which they would be executed during data entry. The report lists the field or screen that causes the rule to execute (trigger field), the field or screen that receives the action (action field or target screen) and the translation. Rules that include a variable that has been deleted from the logic in the module will display the deleted variable as "-".

---

## ANALYSIS REPORTS

---

---

### Analysis Report

---

This report is currently not available.

## Derived Variable Report

---

This report is currently not available.

## Analysis Storyboard (HTML) Report

---

While conducting an analysis it may be desirable to capture a view or graph to share with other analysts via the Internet. The HTML storyboard report will save a view and any notes associated with an indicator in .HTM file format and a graph in a JPEG file format. You can view this type of report in any HTML browser.

---

# HELP

---

---

## GENERAL HELP

---

The **General Help** provides general MedQuest Analyzer system help.

---

## MODULE HELP

---

If you have compiled the clinical help (in MedQuest) entered in your module, then you can view that help file (XXXDICT.HLP where XXX is the module acronym) here.

---

## ABOUT SOFTWARE

---

**About Help** identifies the MedQuest Analyzer and dictionary version information.

---

# APPENDIX A

---

---

## DESIGN FUNCTIONS

---

---

### Add a Variable

---

A single variable is the variable that is most common in the design of a data entry system. There are a number of different types of variables that are described below. You must complete all of the required information for a given variable type before MedQuest will save those changes.

---

#### Variable Identification

---

**NAME.** You must enter a unique name (8 characters or less) to identify each variable. You are allowed to reuse a variable you have already defined (duplicate variable) on different data entry screens. If you type in the name of a variable that has already been used, you will be prompted with a message asking whether you want to retrieve that variable and its properties.

You can also use the **Quick Copy** feature to duplicate an existing variable in your current module. This feature requires that you select a screen set, screen, and the variable to quick copy. If you want to copy a variable from another module, you should use the **Query/Copy** function.

#### NOTE:

Be careful if you reuse a variable but do not copy its properties into your **Design** window. When you save your changes, whatever changes you make will be saved for that variable name wherever it is used.



#### Advanced User: Naming Strategies

It is recommended you begin each new variable name with the three-letter module acronym. This convention allows for easier manipulation of the variable during other phases of the project, particularly the data analysis phase.

**SHORT TITLE.** The short title (30 characters or less) is a short description of the variable that is used primarily in reports and places where space is limited. For example, the short title is used in the **Case Summary** report and is used by the MedQuest Quality (IQC) program to display the differences between two cases entered for a module. It is advantageous to have a meaningful short title.

**SCREEN TITLE.** The long title (254 characters or less) is what appears on the **Data Entry** screen. It is recommended that you make this as concise as possible to leave enough space on the screen for other variables.

---

#### Variable Type

---

Following are the types of variables that are available and the properties you are required to enter for each type:

**OPTION (PICK ONE).** This variable type enables you to define a list of options from which the data abstractor can select only one option. Each option (64 characters or less) can be added, deleted, changed, or moved on the list. You are allowed to enter up to 50 options (25 or fewer are recommended).

A set of default options (i.e., Yes/No, True/False, None of Above, UTD) are also provided as commonly used options. By default, the options are assigned a value to be saved (e.g., first option = 1, second option = 2, etc.) in data entry. If you want to change that value, you can do so. Each value that is saved can only be one or two characters long. Note that the default options you can select have default values that are saved for that option.

It is recommended that you keep your options short to minimize the impact of the available space on the screen.

#### **EXAMPLE:**

Use Option (Pick One) to collect information on whether the patient is Male or Female. Assign "1" as a value to indicate male, "2" to indicate female, and "0" to indicate UTD.

**OPTION (PICK ONE OR MORE).** This variable type enables you to define a list of options from which the data abstractor can select more than one option. You will have to define a list of options (each one 64 characters or less) to be selected. The last item on the list will clear all of those above it during data entry and is typically the None Of Above option. You can create up to 50 options (25 or fewer are recommended).



#### **Advanced User: How Data Are Saved**

In data entry, each option type is saved as a 0 (not marked) or a 1 (marked) in a string. The first element in this string is always associated with the last option to allow easy analysis of the option that clears all of the others. The first option is saved in the second position in the string, etc.

#### **EXAMPLE:**

You have Option 1, Option 2, Option 3, and None of Above. If the option None of Above is selected, the other options will be cleared and the string saved will be "1000." Selecting Option 1 and Option 3 will save the string as "0101." Selecting Option 2 will save the string as "0010." Selecting all of the options (except None of Above) will save the string as "0111."

**OPTION (EXTERNAL LIST).** You should select this option when you have a list of options that exceeds the MedQuest limitation for Option (Pick one) or Option (Pick one or more). This requires you to create a Microsoft Access database containing the list of options and the value you want saved. When you select this option, you will be required to provide the external database name, the table in that database that contains your options, the name of the field in that table that contains the options to display on the screen, and the name of the field in that table that contains the value to save. You will also be required to enter the number of selections the user is allowed to make.

When you design the database table, you must meet the following specifications:

- Field to Save field length =< 10 characters
- Field to Save field must have a unique value for each record
- Index with same name as Field to Save field must exist on the field
- Index with same name as Field to Display field must exist on the field
- Database table has 500 or fewer records

### EXAMPLE:

Use this variable type when you have a list of microorganisms in a database from which you want the data abstractor to pick.



### Advanced User: Design Strategy

When the data are saved, each option the data abstractor selects has the **Field to Save** value saved; each of these values is separated from the other by a semicolon. For example, if you are saving three options that have values 001, 024 and 099, the saved string would be "001;024;099;". Since the maximum length of the **Field to Save** is 254 characters, you must strategically design your external database if you are going to allow the user to select more than one option. The system will calculate the maximum number of options by multiplying the number of allowed options by the length of the **Field to Save** plus 1. If this result exceeds 254, a warning is given that too many options are being allowed.

It is recommended that the **Field to Save** field length be 2 or 3 characters. Another good design practice is to keep the number of options in the database to 400 or less. The current design loads all of the options on the list; therefore, the longer the list, the more time it takes to load.

It is also recommended that you create the external database table in your module dictionary (XXXDICT.MDB) to simplify the copying/transferring of dictionaries.

**DATE.** This variable type requires that you select the format in which you want the date to be entered. There is a list of allowed formats from which you can select. These formats apply only to data entry. When the data are saved, all dates without a century requirement are saved as MM/DD/YY. All dates with a century requirement are saved as MM/DD/YYYY.

**TIME.** This variable type requires that you select the format in which you want the time to be entered. There is a list of allowed formats from which you can select. These formats apply only to data entry. When the data are saved, times are saved as HH:MM if no seconds are included or as HH:MM:SS if seconds are included in the format.

**DATE/TIME.** This variable type requires that you select the format for both date and time in which you want the date/time to be entered. There is a list of allowed formats from which you can select. These formats apply only to data entry. When the data are saved, if the format requires a century and seconds, the data will be saved as MM/DD/YYYY HH:MM:SS, otherwise it will be saved as MM/DD/YY HH:MM.

**STRING.** This variable type enables the data abstractor to enter any string of characters. The designer is required to enter the length of the string up to 99 characters. A feature "Require all digits," can be selected when creating this type of variable to force entry of all of the digits.

### EXAMPLE:

Social security numbers are nine digits long. If this option has been selected, the data abstractor must enter all nine digits.

**MEMO.** This variable type allows the user to enter up to 32,000 alphanumeric characters. Since the memo data can be lengthy, MedQuest provides an Import function that allows the data abstractor to import a text file into the memo field.

### NOTE:

Since this is a free form text field, the data collected will not be conducive for analysis.

**NUMBER (NO UNITS).** The number with no units requires you to specify the maximum number of whole digits and decimal place digits that the data abstractor is allowed to enter. During entry of a number, the

data abstractor will be allowed to enter only one decimal place at any location within the number. A feature "Require all digits," can be selected when creating this type of variable to force entry of all of the digits.

**NUMBER (UNITS).** For a number with units, you must specify the maximum number of whole digits and decimal place digits that the data abstractor is allowed to enter. You must also select at least two units from a group of units listed. If you need a new unit group to select from, you can use the Edit Units option under the UTILITIES menu to define it.

The first unit in the list serves as the base unit in which the data are saved. For example, if you specify Pounds and Kilograms as the two units, the data will be saved in Pounds. If you specify Kilograms and Pounds as the two units, the data will be saved in Kilograms.

If you select "UTD" as a unit, the data abstractor may enter a number with unknown units. When the value is saved, a "U" is appended to the end of the string.

A feature "Require all digits," can be selected when creating this type of variable to force entry of all of the digits.

#### **EXAMPLE:**

Use variable type Number with units if you want the data abstractor to enter a patient's height in centimeters or enter the height in inches and have the value converted to centimeters.

**NUMBER SET.** A number set requires you to specify the maximum number of whole digits and decimal place digits that the data abstractor is allowed to enter. In addition, you can define up to six options that will appear on the screen.

A number set is a set of options (typically units of measurement) where you can enter a number for each option in the list. For example, you might need to collect the number of years, months and days since a patient was last pregnant. In this case, the options you would enter would be years, months and days.

When the data are saved, each value is saved (in the order of the options) in a string separated by a semicolon. For example, 3 years, 2 months and 1 day would be saved as "3;2;1;".

A feature "Require all digits," can be selected when creating this type of variable to force entry of all of the digits.



#### **Advanced User: Maximum Number Of Options**

Since you are limited to saving 254 characters of data, the maximum number of digits (whole plus decimal plus 1(for the semicolon)) multiplied by the number of options must be less than 254 characters. If it exceeds 254, you will be given a warning.

**ICD9 CODE.** This variable type provides the data abstractor with ICD-9-CM diagnosis codes (and descriptions) and ICD-9-CM procedure codes (and descriptions) from which the selection(s) can be made. During the design you are required to select the type of code (i.e. ICD-9-CM diagnosis or ICD-9-CM procedure codes) that you want to collect during data entry.

**MEMO (ICD9).** This variable type allows the data abstractor to enter up to 32,000 characters. In addition, the ability to retrieve and embed an ICD-9-CM diagnosis code (and description) or an ICD-9-CM procedure code (and description) is provided. You can select the type of code you want to provide from the **Edit Variable** window. During data entry, the data abstractor will be provided with an assistive device that allows him/her to retrieve the codes. Since the memo data can be lengthy, MedQuest provides an **Import** function that allows the data abstractor to import a text file into the memo field.

**NOTE:**

Since this is a free form text field, the data collected will not be conducive for analysis.

**MEMO (LIST OF).** This variable type allows the data abstractor to enter data for each item in a list of memos. You must add one or more memo titles of the data in the list of memos provided. The abstractor can enter a total of 32,000 characters for all of the items in the list combined. Since the memo data can be lengthy, MedQuest provides an **Import** function that allows the data abstractor to import a text file into the memo field.

**NOTE:**

Since this is a free form text field, the data collected will not be conducive for analysis.

**DIAGRAM.** This variable type allows the data abstractor to mark up one or more diagrams each identified by the diagram title. You must add one or more diagram titles for the diagrams to be collected in the list provided. In addition, for each diagram, you must identify the bitmap file that is to be used. Since the user can also enter remarks about the diagram and these data can be lengthy, MedQuest provides an **Import** function that allows the data abstractor to import a text file into the memo field.

During data entry, the data abstractor will be able to mark up the bitmap for each case with lines, circles, arrows and text. In addition, he/she will be able to write comments for each of the diagrams.

**NOTE:**

Since this is a free form text field, the data collected will not be conducive for analysis.

**LABEL.** This variable type does not require data entry. It is used to place a set of text on the screen that is not associated with any data entry. You are not required to enter any information except the variable name, short title and screen title. Data are not saved for this variable type.

**Advanced User: Not Sure Of Variable Type**

If you are designing a variable and you do not know its variable type, you can create a label as a place holder and then later refine the variable by switching it to the applicable type.

## Variable Options

---

**MANDATORY.** When a variable is marked as Mandatory, it means that on the **Post Processor** screen (if you have turned on post processing after case entry and/or post processing after screen entry), an error message will be given (if no entry exists for the variable) that the variable is mandatory and must be entered before the case can be marked as complete.

**NOTE:**

The default for each variable is always mandatory. The mandatory flag applies the same way to all of the duplicate variables.

**READ ONLY.** When a variable is marked as **Read Only**, the data abstractor will not be able to enter any data for that variable on the given screen during data entry.

**EXAMPLE:**

If you want the data abstractor to enter the variable **Discharge date** on the **Demographics** screen, but only view this variable on the **Discharge** screen, you can mark the variable **Discharge date** on the



**Discharge** screen as **Read Only**. (Hint: remember, you can use the same variable on more than one screen).

**DERIVED**. Within the analysis tool the only type of variable you can add is a derived variable.

**DEFAULT VALUE**. When a variable is marked as **Default Value**, you will be required to enter a default value for that variable. If the variable is marked **Mandatory**, the data abstractor will be able to change the default value during data entry. If the variable is marked **Read Only**, the data abstractor will not be able to change the default value. The default values are loaded when you create a new case or when you add a record to a grid.

**EXAMPLE:**

If you create a date variable and choose the Default Value option, you will be required to save a date (e.g., 01/01/96) value for that variable.

---

## QUERY/ADD VARIABLES

---

You can query for variables in other modules to retrieve and add them to the current **Design** window. This function can also be used just to query other variables in other modules (e.g., to quickly view the help for a variable) and to retrieve (add) help for another variable to the workspace where you are editing help. The objective of the general query interface is to provide you common access to variables in other modules.

**PATH TO LIST OF MODULES**. You can point to any project file (which lists the modules and the locations of their dictionaries) by changing the directory path in the box provided. By default, the path is set to the current path you are using in MedQuest.

**SEARCH MECHANISMS (QUERY CRITERIA)**. There are three ways you can search for variable(s). The first option, by Screen, lets you select a module and a screen in that module. When you select the screen, all of the variables on that screen are placed in the query result list.

Second, you can search by any word(s). You can perform this search on screen sets, screens or variables by selecting the type of query. To complete the query, you must select the module you want to query, and then enter the word, string or phrase for which you are searching into the boxes provided. Currently, you can search by name (variable only), short title, screen title, or help. These are the fields that are searched in the dictionary. You can enter combinations of words which result in an "AND" search string. When you select the **Query** function on the screen, the variables or screens that match the search criteria will be placed in the *Query Result* box.

Finally, you can search by keyword. These keywords are the same keywords that you have built using the **Edit Keywords** function in the utilities. You accomplish this by selecting the module to search and then typing in the keyword that you want to search. Note that as you type in a key word an alphabetical search is performed and all keywords that match your string are displayed plus the next 50 items in the keyword list. When you select a keyword from the list and select the **Query** function, the variables that are classified by the keyword will be placed in the query result list.

**QUERY RESULTS**. Once you have run your query, the list of variables matching your query will be displayed on the query result list.

When you select an item in the query result list, any help that you have entered for that item will appear in the help box provided. This makes it easy for you to see the help for other variables in other systems.

For variables in the query result list, you can view additional details about a variable by selecting the **Preview** function. This provides a more English-like description of the variable.

**ADD/COPY VARIABLES/HELP.** If you are using the **Query Variables** window to add variables to your current **Design** window, you can do so by selecting all of the variables in the query result list that you want to add, and then select the **Add** function.

If the **Query Variables** window is being used to retrieve a variable's help into the current **Edit Variable Help** window, you should select one variable (you are only allowed to select one) and then select the **Add** function.

If you are using the **Query Variables** window to copy a variable's properties to your variable in the **Design** window, then you should select one variable and select the **Copy** function.

---

# GLOSSARY

---

---

## LIST OF TERMS

---

### ANALYSIS

An analysis is a group of related items used to answer questions about the clinical data. Once the data are collected, you want to begin analyzing them. This analysis process is typically an open ended approach that may take different paths depending on the information collected, e.g., you may perform statistical analysis on the data or use a commercial reporting tool to produce a set of reports.

Often, the analysis process involves the construction of a set of questions or queries of the data. The results of this analysis produces a subset of the raw data that can be easily represented. This process of query building is what the analysis tool was built to support since it is a relatively straightforward and simple process compared to more complex steps of analysis.

### DATA ENTRY

A MedQuest function that allows you to collect data for the data entry system designed by the MedQuest Design function by creating the data store file called **XXX.MDB** where "XXX" represents the module acronym (e.g., PNE.MDB is the data store file for the data entry module Pneumonia).

### DATA ENTRY RULE

A rule specified during the design process that is executed during data entry.

### DATA ENTRY SYSTEM (DES)

A system designed by MedQuest to collect clinical data for a data analysis project. Each project is called a Module and is represented by a three-character acronym (e.g., PNE is the Data Entry System for the Pneumonia project).

### DERIVED VARIABLES

Often in analysis, you may find it necessary to use the raw data to infer a given result, i.e., whether a given clinical event such as an MI took place. The **MedQuest Analysis Tool** lets you do this by building a derived variable. You can build a derived variable by building an equation that describes the result. For example, you can build a derived rule that says "If ... and ... and ... then MI=true."

In the analysis process, this stage of development is laying the foundation for performing more complex queries by breaking the parts of the complex queries into smaller pieces.

### DESIGN

A MedQuest function that allows you to develop a data dictionary for a data entry system by creating a file called **XXXDICT.MDB** where "XXX" represents the module acronym (e.g., PNEDICT.MDB is the data dictionary file for the data entry module Pneumonia).

## **GRAPHS**

When you are working with an INDICATOR or a VIEW, it is often useful to present the data in a graphical format. The MedQuest Analyzer lets you create any number of graphs (based on the variations in the data) for both the indicator logic or view logic.

## **INDICATORS**

In the MedQuest Analyzer, you can build your query of the data by creating an **Indicator**. When you create (or edit) an indicator, you are selecting the variables that you want in your query. The analysis tool will generate an SQL statement after the indicator (query) building template is completed.

To build a query (in equation form), options are provided to describe the records to be included or excluded, the variables to be displayed, the variables to be grouped by (and filtered for), and the variables to be ordered by. In addition, the capability is provided to identify variables to be included in dynamic queries during the analysis, e.g., you may want to vary the diagnosis code that is included (or excluded) as the data are analyzed.

An indicator will produce a data table that is displayed and that can also be graphically represented.

## **MEDICAL DATA ENTRY DESIGN SYSTEM (MEDQUEST)**

An application used for designing a data entry system and collecting data for that system.

## **MODULE**

See Data Entry System.

## **MODULE VARIABLES**

Module variables are variables created by the designer. They are specific to the Module being developed or used.

## **QUALITY CONTROL DATA**

As you collect the data, you can perform quality control (if you are doing retrospective data collection) by using the **MedQuest Quality (IQC)** utility. This tool allows for the same case to be abstracted by different users and has a mechanism for comparing the results and pointing out quality control problems. In addition, you can build a series of quality control queries using the **MedQuest Analyzer** (or other tools such as Microsoft Access) that can identify quality control problems by searching for certain patterns in the data (e.g., data out of range).

## **SCREENS**

A screen is an area beneath the Tab where variables belonging to the same type are laid out (e.g., Tab **Laboratory** represents the Laboratory screen and collects laboratory-related variables). A Tab represents a screen. The variables on the screen are displayed by selecting a Tab. A Tab may contain one or more Subtabs.

## **SCRIPTS**

In performing an analysis, you will be creating and modifying the three basic analysis objects: Indicator; View; and Graph. You may want to organize these different sets of objects into different sequences to present or explain the results of your analysis. This sequencing can be accomplished using the SCRIPT object. In this object type, you can add any indicator, view or graph that is a part of

the given analysis to the script and then allow the user to "play back" the script in the sequence you constructed.

The MedQuest Analysis Tool provides you with the capability to begin building a "data warehouse" of clinical data that can be queried and the results presented in tabular or graphical format. These data can then be used with other statistical packages to perform more complex analysis. The MedQuest Analysis Tool supports this next step by enabling you to export the data you have constructed for use with other analytical software.

## **SET OF SCREENS**

The group or groups of screens belonging to a selected DES. If the data entry system to be created or being edited covers a narrowly-defined subject area and contains a reasonable number of variables, only one screen set, the **General Screens**, is necessary.

If the DES being created or being edited covers a discipline or a complex subject that can be logically divided into discrete subject areas, each of these subject areas can be represented by a set of screens (e.g., **Lumbar Disc Disease**, **Head Injury**, and **Brain Tumor** are three sets of screens belonging to the DES called Neurology).

## **SUBSCREENS**

A subscreen is an area beneath the Subtab where variables belonging to the same type are laid out. A Subtab represents a subscreen. The variables on the subscreen are displayed by selecting a Subtab.

## **TAB/SUBTAB**

A Tab/Subtab indicates the screen labels (e.g., Tab **History** is the tab that is used for retrieving the **History** screen).

## **VARIABLE TYPES**

Variable types and attributes determine the type of data the system will accept (e.g., a variable will accept date format, etc.) during the data entry. The designer defines a variable type for each variable. For example, if variable **Admission Date** accepts only data that are in date form, the variable type must be defined as variable type **Date**.

## **VARIABLES**

A variable is a data entry field that accepts data according to the specifications and rules indicated by the designer during the design.

## **VIEWS**

As you are analyzing the results of an indicator (the SQL query has been executed), you may want to make "minor" variations in the logic to see how the results change. For example, you may want to analyze the change in results for age range < 65 compared to age range > 70. In the MedQuest Analyzer, this process of modifying the indicator (query) logic is called creating a view. The MedQuest Analyzer enables you to dynamically modify the records to be included/excluded and also the order of the data. When you observe a view that you want to keep, you can save this view and it will be included under the indicator.